



**Departamento de Lenguajes y Sistemas Informáticos**  
**E.T.S. Ingeniería Informática. Universidad de Sevilla**

Avda Reina Mercedes s/n. 41012 Sevilla  
Tlf/Fax 954 557 139 E-mail [lsi@lsi.us.es](mailto:lsi@lsi.us.es) Web [www.lsi.us.es](http://www.lsi.us.es)



# **Diseño de bases de datos**

## **Módulo-I Tema 3**

### **Modelo Entidad-Relación Extendido (EE/R)**

Sevilla, febrero 2007  
V 2007.02.1

# Indice

<b>1</b>	<b>INTRODUCCIÓN</b> .....	<b>3</b>
<b>2</b>	<b>ENTIDADES DÉBILES</b> .....	<b>3</b>
2.1	ENTIDAD DÉBIL EN EXISTENCIA.....	3
2.2	ENTIDAD DÉBIL EN IDENTIFICACIÓN.....	3
2.3	SUBCLASE/SUPERCLASE.....	4
2.4	DIAGRAMAS EE/R.....	4
2.5	HERENCIA DE ATRIBUTOS.....	4
2.6	ESPECIALIZACIÓN.....	4
2.7	GENERALIZACIÓN.....	5
2.8	CARACTERIZACIÓN DE LA ESPECIFICACIÓN/GENERALIZACIÓN.....	5
2.9	JERARQUÍA DE GENERALIZACIÓN/ESPECIALIZACIÓN.....	7
<b>3</b>	<b>AGREGACIÓN Y ASOCIACIÓN</b> .....	<b>8</b>
<b>4</b>	<b>MAPEO EE/R A RM/B</b> .....	<b>10</b>
4.1	HEURÍSTICA DE MAPEO EE/R A RM/B.....	10
4.2	COMPARACIÓN DEL MAPEO DE JERARQUÍAS.....	13
<b>5</b>	<b>CASO: EJEMPLO EE/R Y MAPEO A RM/B</b> .....	<b>14</b>

## 1 Introducción

El modelo Entidad-Relación extendido (**EE/R**) incluye los conceptos del modelo Entidad-Relación original (*Peter Chen, 1976*), e incorpora nuevos conceptos que permiten representar requerimientos más complejos.

Diversos autores (*Batini, Markowitz, Shoshani, Teorey, Yang, Fry, ...*) incorporan extensiones semánticas al modelo original. Se estudiarán en este capítulo las características de las extensiones propuestas en el modelo OOER de Navathé-Pillalamarri, incluidas en la referencia bibliográfica del programa (*[ElMasri-Navathé2004] Fundamentos de Bases de Datos: Elmasri-Navathé, 4ª Edición, 2004 Addison Wesley*).

Los modelos EE/R suelen restringir la asociación de atributos a las interrelaciones, restringiendo dicha asignación sólo a entidades y a nuevas primitivas semánticas.

Se estudiarán los conceptos de **subclase/superclase** para representar **jerarquías de generalización/especialización**, así como las primitivas semánticas **agregación** y **asociación**.

## 2 Entidades débiles

### 2.1 Entidad débil en existencia

Una entidad débil se representa con un rectángulo con doble trazo.

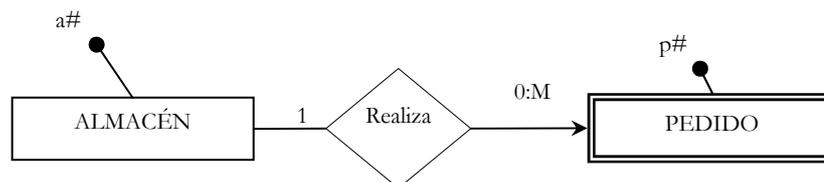


Fig. 1. Entidad débil en existencia

Toda entidad débil (weak entity) es débil en existencia. Ej.: en un almacén (**a#**) se realizan distintos pedidos y un pedido queda identificado por el código de pedido (**p#**). La entidad pedido es débil en existencia.

### 2.2 Entidad débil en identificación

Una entidad débil en identificación, además, se representa con un rombo con doble trazo.

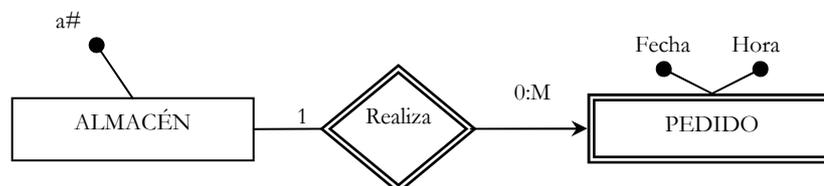


Fig. 2. Entidad débil en identificación

Toda entidad débil en identificación también lo es en existencia. La diferencia estriba en que los atributos de la entidad débil no son suficientes para identificarla.

Ej.: en un almacén (**a#**) se realizan distintos pedidos y un pedido queda identificado por (**fecha, hora y a#**) puesto que en diferentes almacenes podrían darse pedidos concurrentes en el tiempo.

### 2.3 Subclase/superclase

En el modelo ER una entidad tipo representa un conjunto de entidades del mismo tipo; ej. la entidad tipo TÉCNICO.

Un TÉCNICO puede ser por ej. ANALISTA, PROGRAMADOR, CONSULTOR, etc. Tienen en común que todos son TÉCNICOS pero tienen propiedades distintas (atributos) e interrelaciones adicionales con otras entidades según sean ANALISTA, PROGRAMADOR, CONSULTOR, etc.

Es decir una entidad tipo puede tener subagrupaciones de entidades que es importante representar. Cada una de estas subagrupaciones (ANALISTA, PROGRAMADOR, CONSULTOR) es una **subclase** de la entidad TÉCNICO. TÉCNICO es una **superclase**.

Una entidad de la subclase es la misma que la de la superclase pero con un papel específico.

Toda ocurrencia de alguna subclase pertenece a la superclase y no al revés. Es decir todo ANALISTA es un TÉCNICO y no todo TÉCNICO es ANALISTA.

### 2.4 Diagramas EE/R

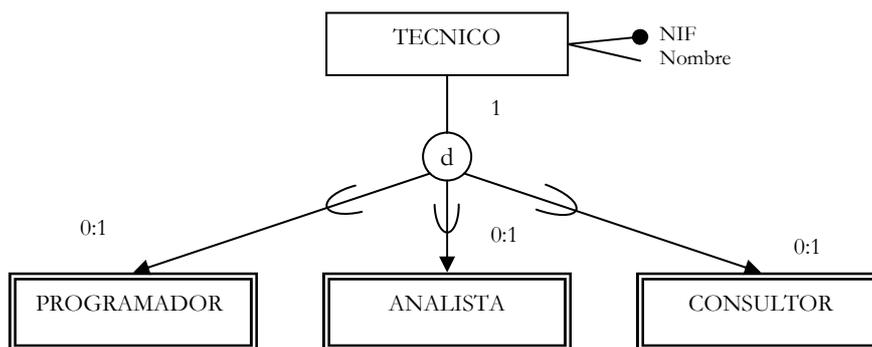


Fig. 3. Subclases y superclases.

### 2.5 Herencia de atributos

Una entidad que es miembro de una subclase hereda todos los atributos de la superclase. Por tanto la subclase posee sus atributos específicos más los que hereda como miembro de la superclase.

### 2.6 Especialización

Es el proceso de definir un conjunto de subclases a partir de una entidad tipo (superclase). Así el conjunto de subclases {ANALISTA, PROGRAMADOR, CONSULTOR} es una especialización de la superclase TÉCNICO.

Pueden existir varias especializaciones de una misma entidad tipo.

La especialización permite

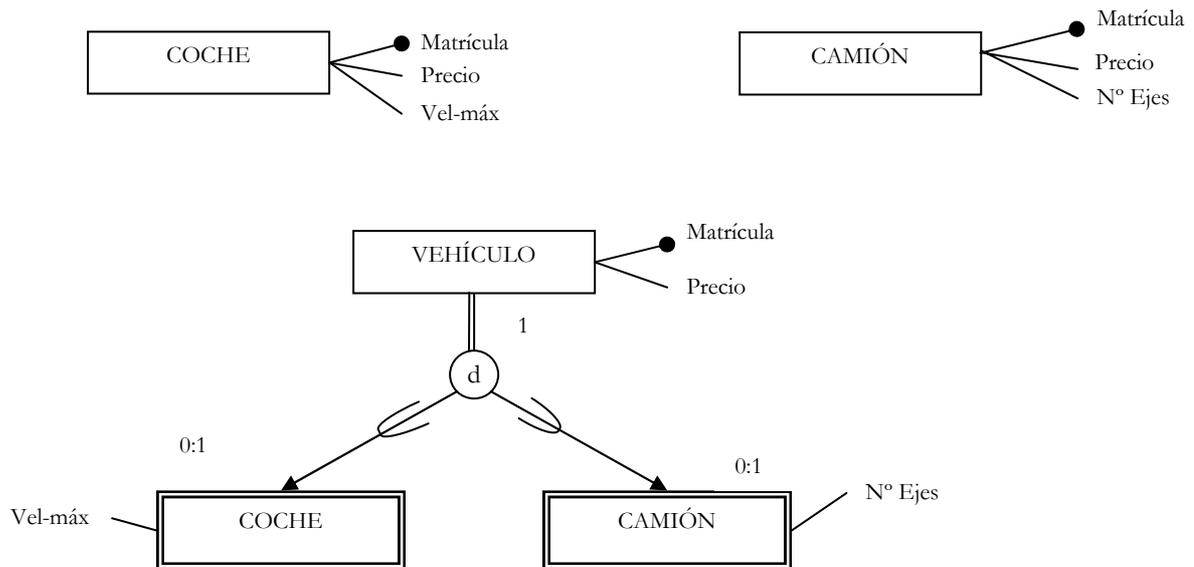
- Asociar atributos específicos a cada subclase.
- Establecer interrelaciones adicionales entre las subclases y otras entidades.

## 2.7 Generalización

Es el proceso de abstracción inverso a la especialización.

Se suprimen las diferencias entre varios tipos de entidades y generalizamos sus características comunes para formar una entidad superclase.

Ej. Tenemos las entidades COCHE y CAMIÓN.



**Fig. 4.** Generalización a partir de las entidades COCHE y CAMIÓN.

Mediante un proceso de generalización obtenemos la entidad VEHÍCULO.

El proceso inverso consiste en ver COCHE y CAMIÓN como una especialización de la superclase VEHÍCULO.

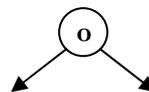
## 2.8 Caracterización de la especificación/generalización

Dependiendo de que las ocurrencias de alguna subclase puedan aparecer o no en más de una subclase podemos diferenciar entre:

- Subclases disjuntas.
- Subclases solapadas.

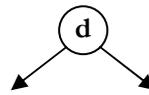
Cuando una ocurrencia de la superclase puede aparecer en más de una subclase decimos que las subclases son **solapadas**.

Se representa por la letra **o** (overlapping) :



Cuando una ocurrencia de la superclase solo aparece en una única subclase decimos que las subclases son **disjuntas**.

Se representa por la letra **d**(disjoint) :

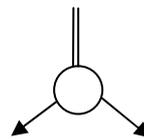


Atendiendo al nivel de recubrimiento de la población (ejemplares/ocurrencias) de la superclase respecto a la población de las superclases pueden representarse dos tipos de jerarquías:

- Jerarquía total
- Jerarquía parcial

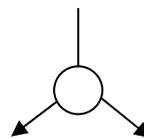
Cuando toda ocurrencia de la superclase aparece al menos en una subclase se dice que la **jerarquía** es **total**.

Se representa por:



Cuando en la superclase existen ocurrencias que no aparecen en ninguna de las subclases se dice que la **jerarquía** es **parcial**.

Se representa por:



Ej. Todas las piezas se compran o fabrican y una pieza puede ser comprada y fabricada

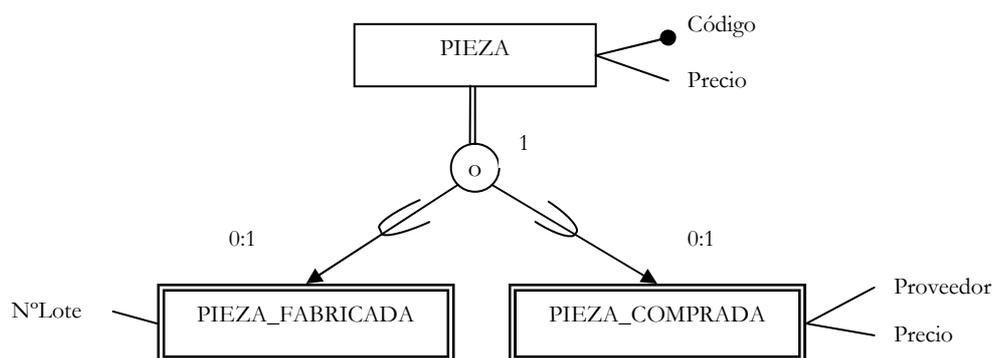


Fig. 5. Jerarquía total con solapamiento

Existen dos razones principales que aconsejan la definición de superclases y subclases:

- Hay atributos que no pueden ser aplicables a todas las ocurrencias de la entidad. Por ej. sólo nos interesa saber los años de experiencia de los consultores y no de los analistas y programadores.
- Algunas ocurrencias de la entidad tienen interrelaciones adicionales con otras entidades. Por ej. sólo los analistas dirigen proyectos (sólo los analistas se interrelacionan con los proyectos).

## 2.9 Jerarquía de generalización/especialización

Es posible especificar subclases de una subclase formando una jerarquía de generalización/especialización:

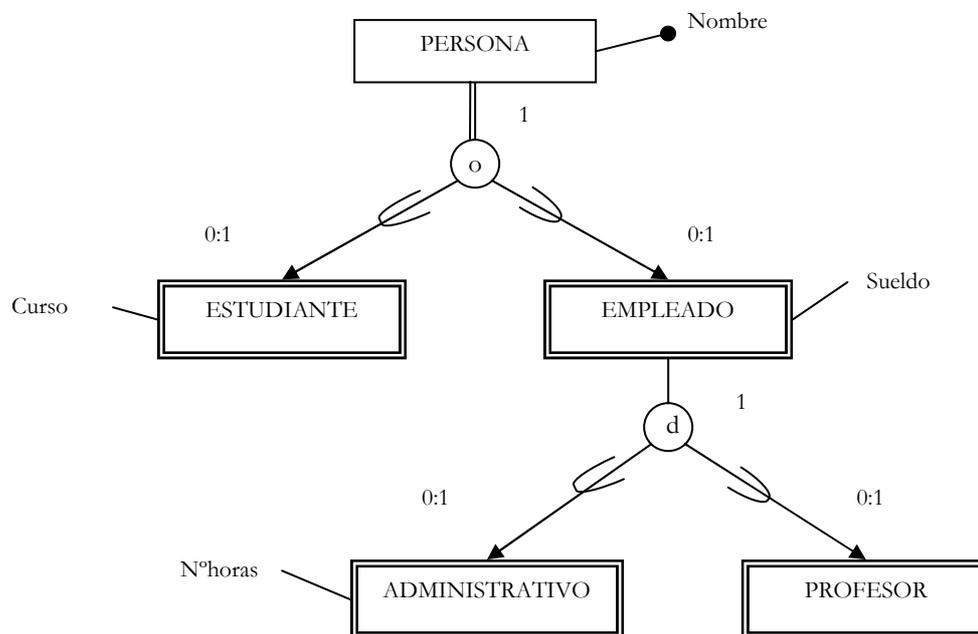


Fig 6. Jerarquía de especialización

EMPLEADO es una subclase de PERSONA y es también una superclase de {ADMINISTRATIVO, PROFESOR}. Esto implica la restricción de que todo profesor ha de ser un empleado.

Una subclase puede ser subclase en más de un vínculo clase/subclase.

Ej. JEFE\_PROYECTO es una subclase de CONSULTOR y de ASALARIADO. Por tanto para ser JEFE\_PROYECTO es necesario ser CONSULTOR y ASALARIADO. JEFE\_PROYECTO es una subclase compartida.

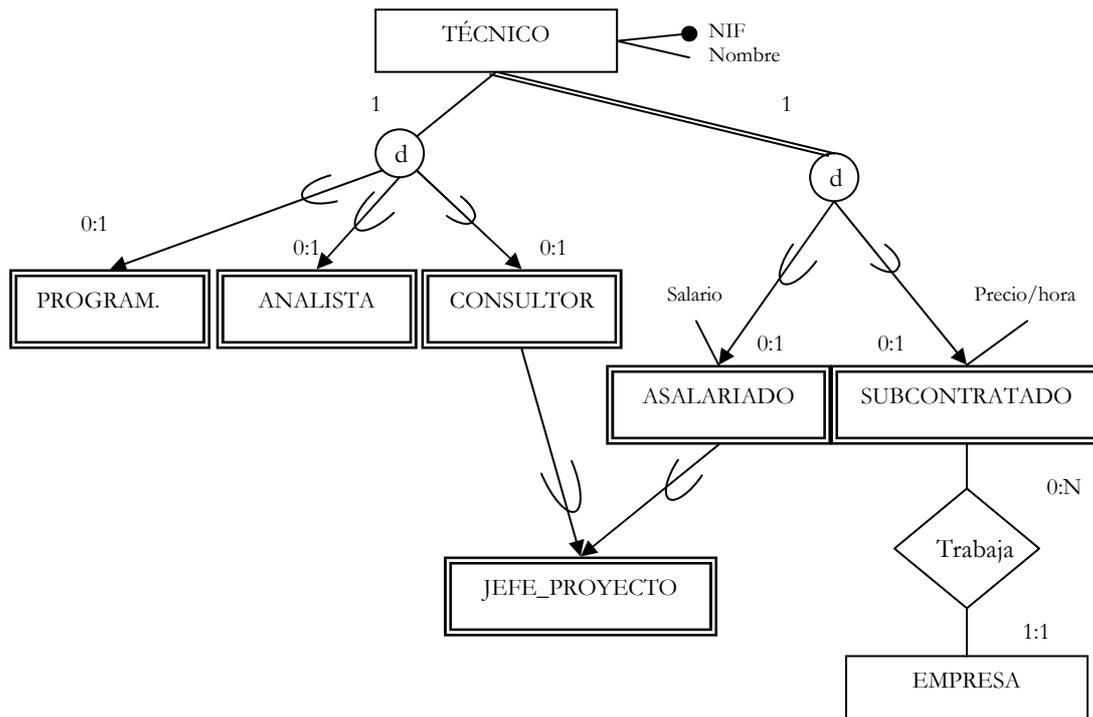


Fig. 7. Retícula de generalización/especialización

### 3 Agregación y Asociación

La **agregación** es un concepto de abstracción para construir objetos compuestos a partir de sus objetos componentes.

Permite combinar entidades entre las que existe una interrelación y formar una entidad de más alto nivel. Es útil cuando la entidad de más alto nivel se tiene que interrelacionar con otra entidad

Ej. Un técnico puede trabajar en varios proyectos y en un proyecto trabajan varios técnicos. Como consecuencia del trabajo de un técnico en un proyecto puede publicar uno o varios artículos.

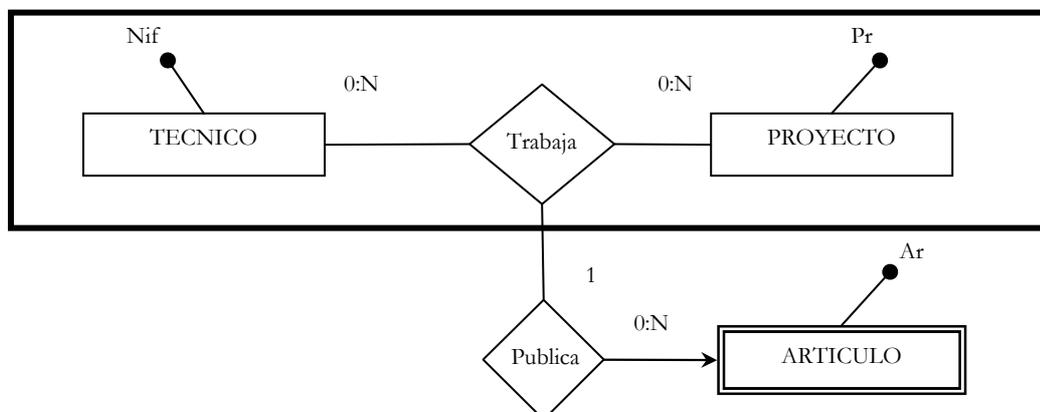


Fig. 8. Agregación.

La abstracción de **asociación** permite asociar o vincular dos entidades independientes.

Una **asociación** queda identificada por la identificación de las entidades participantes.

Una diferencia entre asociación y agregación es que al eliminar la asociación las entidades participantes siguen existiendo. En la agregación si se elimina la entidad agregada se eliminan además las entidades que la forman

La forma de representar la asociación según los autores [Elmasri/Navathé2004] consiste en crear una nueva entidad TRABAJA que depende en identificación de TECNICO y PROYECTO.

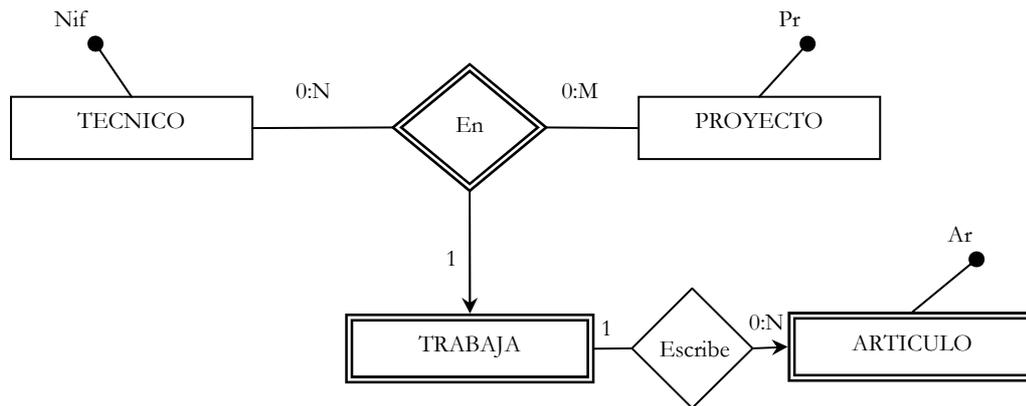


Fig. 9. Representación de la asociación según [Elmasri/Navathé2004]

Autores como Rob P, Coronell C<sup>1</sup> proponen la representación de la asociación como un nuevo tipo de entidad denominada **entidad compuesta**, definiendo un símbolo que combina la representación de una entidad y una interrelación (tiene el comportamiento de ambas). De este modo, una entidad compuesta puede, a su vez, participar en otras interrelaciones en el modelo.

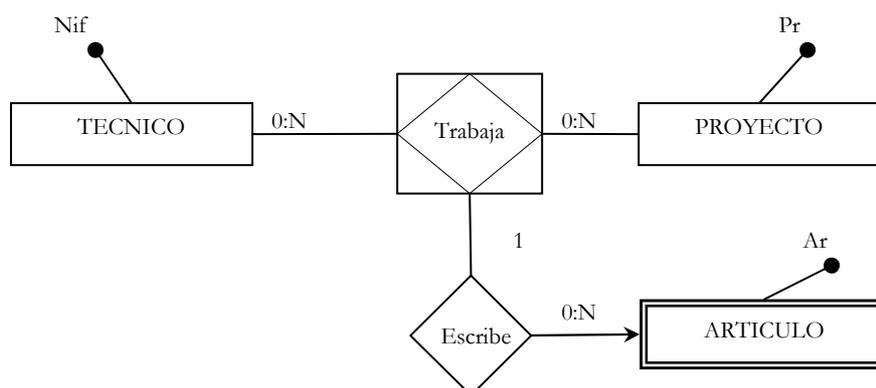


Fig. 10. Representación de una **asociación** como **entidad compuesta**.

En adelante y en los ejercicios propuestos en la asignatura se utilizará esta notación para la asociación o entidad compuesta.

<sup>1</sup> Sistemas de Bases de Datos; Diseño, Implementación y Administración. Rob P, Coronell C, 2004 International Thomson.

## 4 Mapeo EE/R a RM/B

La generación de gestores de bases de datos actuales están todavía basados en el modelo básico (RM/B) de Codd, aunque suelen incorporar algunas extensiones semánticas al mismo (disparadores, encapsulación de funciones, procedimientos, etc.).

En las fases de diseño tecnológico es común la utilización de gramáticas (basadas en estándares SQL o bien diseño propietario) que atienden a este modelo (RM/B); es por tanto necesario (en casos en que la modelación conceptual está basada en modelos EE/R) aplicar métodos que transformen la semántica de un modelo en otro. No debe olvidarse en estas transformaciones la teoría de la normalización (formas normales), aunque esta, en modelos complejos no suele ser directamente aplicable (Métodos de descomposición o de síntesis). Suelen emplearse métodos heurísticos que, a partir de un buen modelo de entidades, generan extensiones adecuadas; es decir: extensiones o esquemas relacionales normalizados.

A continuación se presente la heurística propuesta por [Elmasri/Navathé2004] y se justifica la validez del método o nivel de normalización.

### 4.1 Heurística de mapeo EE/R a RM/B

#### Paso 1: Mapeo de entidades regulares

##### **Caso 1a. Atributos monovaluados.**

Dada una entidad **A**, con identificador  $I_a$ , y con las propiedades  $\{a_1, a_2, \dots, a_n\}$ , se genera una relación  $R_A(I_a, a_1, a_2, \dots, a_n)$  PK( $I_a$ ).

Justificación: Si todos los atributos  $a_i$  son independientes entre sí y monovaluados, entonces puede asegurarse que las únicas dependencias son del tipo  $I_a \rightarrow a_i$ ; por lo que  $R_A$  está en 3FN (ya que  $\forall I_a \rightarrow a_i: I_a$  es superclave).

##### **Caso 1b. Atributos multivaluados.**

Dada una entidad **A**, con identificador  $I_a$ , y con las propiedades multivaluadas  $\{v_1, v_2, \dots, v_n\}$ , se genera una relación  $R_{Avi}(I_a, v_i)$  PK( $I_a, v_i$ ) FK( $I_a$ )/  $R_A$ .

Justificación: Al no haber dependencias funcionales entre  $(I_a, v_i)$ , lo que puede asegurarse por el axioma de reflexividad de Armstrong es  $(I_a, v_i) \rightarrow (I_a, v_i)$  por lo que  $R_A$  está en 3FN (ya que  $\forall (I_a, v_i) \rightarrow (I_a, v_i): (I_a, v_i)$  es superclave).

**Paso 2: Mapeo de entidades débiles****Caso 2a. Débil en existencia.**

Dada una entidad **W**, débil en existencia de la entidad **A** (anteriormente mapeada), con identificador  $I_w$ , y con las propiedades  $\{w_1, w_2, \dots, w_n\}$  se genera una relación

$R_W(I_w, w_1, w_2, \dots, w_n, I_a)$  PK( $I_w$ ) FK( $I_a$ ) /  $R_A$ .

Justificación: Si todos los atributos  $w_i$  son independientes entre sí, se está en la misma situación de referencia de los casos **1a**(monovaluados para entidades fuertes) y **1b**(multivaluados), entonces puede asegurarse que  $R_W$  está en 3FN en cuanto a los atributos  $w_i$ ; como, por otro lado, también puede asegurarse que  $I_w \rightarrow I_a$ , entonces  $R_W$  está en 3FN (ya que  $\forall I_w \rightarrow w_i$  y  $I_w \rightarrow I_a$ , y  $I_w$  es superclave).

**Caso 2b. Débil en identificación.**

Dada una entidad **W**, débil en identificación de la entidad **A** (anteriormente mapeada), con atributo(s) identificador(es)  $I_w$  necesarios para identificar una instancia de **W**, y con las propiedades  $\{w_1, w_2, \dots, w_n\}$  se genera una relación  $R_W(I_w, w_1, w_2, \dots, w_n, I_a)$  PK( $I_a, I_w$ ) FK( $I_a$ ) /  $R_A$ .

Justificación: Si todos los atributos  $w_i$  son independientes entre sí y monovaluados, las únicas dependencias son:  $\forall w_i: (I_w, I_a) \rightarrow w_i$ , entonces  $R_W$  está en 3FN ya que  $(I_w, I_a)$  es superclave.

**Paso 3. Mapeo de interrelaciones 1:N**

Si existe una interrelación **S**(de cardinalidad 1:N) entre las entidades **A** y **B** anteriormente mapeadas mediante relaciones  $R_A(I_a, a_1, a_2, \dots, a_n)$  PK( $I_a$ ) y  $R_B(I_b, b_1, b_2, \dots, b_m)$  PK( $I_b$ ), de modo que a una instancia de **B** corresponde como máximo una instancia de **A**, entonces puede ampliarse (propagarse) el esquema de **B**, incluyendo el identificador  $I_a$  de  $R_A$  en la relación  $R_B$ , quedando la relación  $R_B(I_b, b_1, b_2, \dots, b_m, I_a)$  PK( $I_b$ ) FK( $I_a$ ) /  $R_A$ .

Justificación: En este caso, puede afirmarse que  $I_b \rightarrow I_a$ , luego  $R_B$  sigue estando en 3FN.

**Paso 4. Mapeo de asociaciones (agregaciones o interrelaciones M:N:P)**

Si existe una asociación(o una agregación) **S** entre las entidades **A, B, ..X** anteriormente mapeadas mediante relaciones  $R_A(I_a, a_1, a_2, \dots, a_n)$  PK( $I_a$ ),  $R_B(I_b, b_1, b_2, \dots, b_m)$  PK( $I_b$ ),  $R_X(I_x, x_1, x_2, \dots, x_p)$  PK( $I_x$ ) se genera una relación

$R_S(I_a, I_b, \dots, I_x)$  PK( $I_a, I_b, \dots, I_x$ ) FK1( $I_a$ ) /  $R_A$ , FK1( $I_b$ ) /  $R_B, \dots$  FKx( $I_x$ ) /  $R_x$

Justificación: Si la cardinalidad entre las entidades es múltiple (todas con todas tomadas de dos en dos), entonces puede asegurarse por el axioma de reflexividad de Armstrong es  $(I_a, I_b, \dots, I_x) \rightarrow (I_a, I_b, \dots, I_x)$ , con lo que  $R_S$  está en 3FN.

**Caso 4a. S tiene propiedades adicionales (asociación)  $\{s_1, s_2, \dots, s_r\}$ ;** se añaden entonces los atributos a la generación mapeada, quedando el esquema

$R_S(I_a, I_b, \dots, I_x, s_1, s_2, \dots, s_r)$  PK( $I_a, I_b, \dots, I_x$ ) FK1( $I_a$ ) /  $R_A$ , FK1( $I_b$ ) /  $R_B, \dots$  FKx( $I_x$ ) /  $R_x$

Justificación: Puede afirmarse que  $\forall s_i: (I_a, I_b, \dots, I_x) \rightarrow s_i$ , quedando también  $R_S$  en 3FN.

**Caso 4b. Alguna de las cardinalidades de S es simple (1).**

**Ej.:** Si entre las entidades **A,B,..X** puede afirmarse que dada la pareja de instancias de **(A,B)**, identificadas por la tupla **(I<sub>a</sub>, I<sub>b</sub>)** y a esta pareja solo corresponde, como máximo, una instancia de **X**, identificada por **I<sub>x</sub>**, entonces existe la dependencia funcional **(I<sub>a</sub>, I<sub>b</sub>) → I<sub>x</sub>**, con lo que el esquema se ve reducido (**I<sub>x</sub>** desaparece de la clave primaria, pues está en las mismas condiciones de dependencia que el resto de los atributos s<sub>i</sub>) a:

$$R_s(I_a, I_b, \dots, I_x, s_1, s_2, \dots, s_r) \text{ PK}(I_a, I_b) \text{ FK1}(I_a) / R_A, \text{FK1}(I_b) / R_B, \dots, \text{FK}_x(I_x) / R_x$$

Justificación: Puede afirmarse que  $\forall s_i: (I_a, I_b) \rightarrow s_i$  y que  $(I_a, I_b) \rightarrow I_x$ , quedando también **R<sub>s</sub>** en **3FN**.

**Paso 5. Mapeo de jerarquías de generalización/especialización**

Sea la superclase **P** con propiedades  $\{I, p_1, p_2, \dots, p_m\}$  siendo **I** el atributo identificador de **P** y sean los subtipos  $\{S_i$  con propiedades  $\{s_{i1}, s_{i2}, \dots, s_{ir}\}\}$ , entonces pueden plantearse cuatro alternativas de mapeo:

**Caso 5a.** Se genera una relación para la superclase **R<sub>p</sub>**  $(I, p_1, p_2, \dots, p_m) \text{ PK}(I)$  y una relación para cada una de las subclases **R<sub>i</sub>**  $(I, s_{i1}, s_{i2}, \dots, s_{ir}) \text{ PK}(I) \text{ FK}(I) / R_p$ .

Justificación: Para  $\forall p_m: I \rightarrow p_m$  y para  $\forall s_{ir}: I \rightarrow s_{ir}$ , por lo que **R<sub>p</sub>** y **R<sub>i</sub>** están en **3FN**.

**Caso 5b.** Se genera una relación para cada **S<sub>i</sub>** añadiendo todos los atributos de la superclase:

$$R_i(I, s_{i1}, s_{i2}, \dots, s_{ir}, p_1, p_2, \dots, p_m) \text{ PK}(I).$$

Justificación:  $\forall s_{ir}: I \rightarrow s_{ir}$ ,  $\forall p_m: I \rightarrow p_m$ , por lo que **R<sub>i</sub>** están en **3FN**.

**Caso 5c.** Se genera una única relación con los atributos de **P** y la unión de todos los atributos de los **S<sub>i</sub>** y añadiendo un atributo que defina el subtipo (categoría o **discriminante**):

$$R_p(I, p_1, p_2, \dots, p_m, \text{discriminante}, s_{11}, s_{12}, \dots, s_{1r}, s_{21}, s_{22}, \dots, s_{2r}, \dots, s_{i1}, s_{i2}, \dots, s_{ir}) \text{ PK}(I)$$

Justificación:  $I \rightarrow \text{discriminante}$ ,  $\forall p_m: I \rightarrow p_m$ ,  $\forall s_{ir}: I \rightarrow s_{ir}$ , por lo que **R<sub>p</sub>** está en **3FN**.

**Caso 5d.** Se genera una única relación con los atributos de **P** y la unión de todos los atributos de los **S<sub>i</sub>** y añadiendo un vector (**vector\_cat** (**cat<sub>1</sub>**, **cat<sub>2</sub>**, .., **cat<sub>i</sub>**)) donde **cat<sub>i</sub>** es un booleano que determina si la instancia o ejemplar pertenece a la subclase).

$$R_p(I, p_1, p_2, \dots, p_m, \text{cat}_1, s_{11}, s_{12}, \dots, s_{1r}, \text{cat}_2, s_{21}, s_{22}, \dots, s_{2r}, \dots, \text{cat}_i, s_{i1}, s_{i2}, \dots, s_{ir}) \text{ PK}(I)$$

Justificación:  $\forall \text{cat}_i: I \rightarrow \text{cat}_i$ ,  $\forall p_m: I \rightarrow p_m$ ,  $\forall s_{ir}: I \rightarrow s_{ir}$ , por lo que **R<sub>p</sub>** está en **3FN**.

## 4.2 Comparación del mapeo de jerarquías

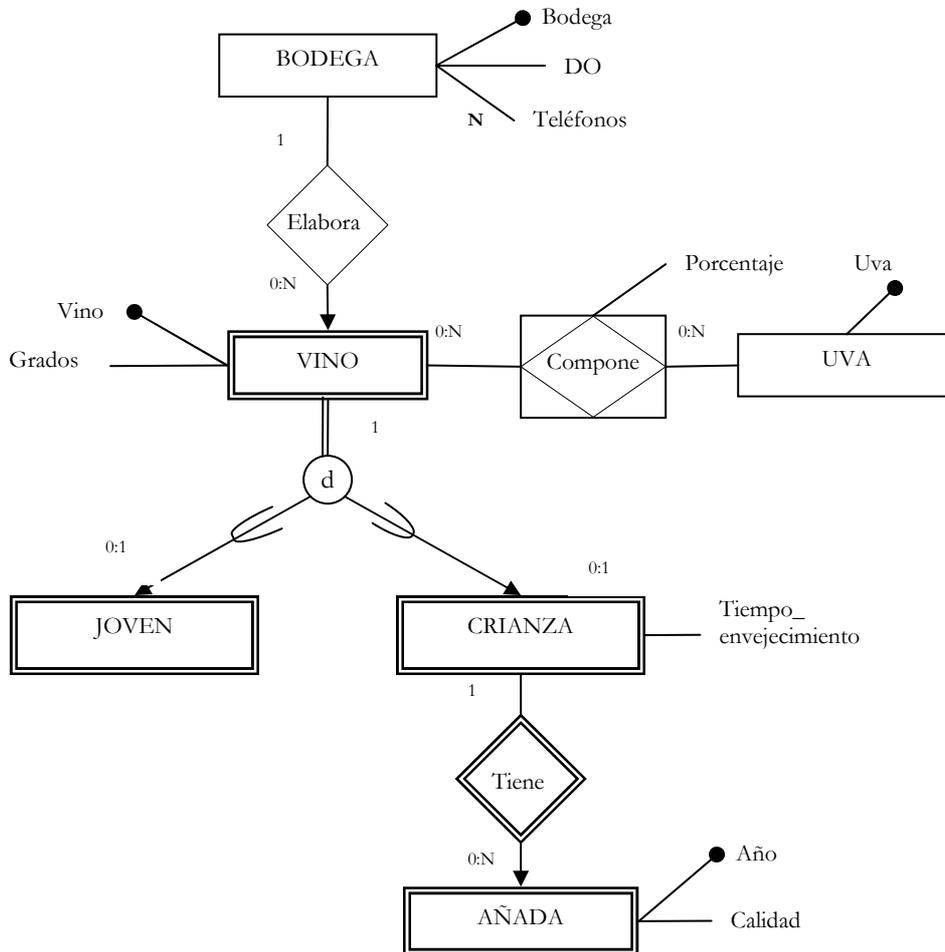
Se presenta a continuación un cuadro comparativo del comportamiento de las opciones de mapeo:

Paso	Relaciones creadas	Comportamiento de la jerarquía				Nulos
		Parcial	Total	Disjuntas	Solapadas	
5a	$R_p, \{R_1, R_2, \dots, R_i\}$	Bueno	Bueno	Bueno	Bueno	Bueno
5b	$\{R_1, R_2, \dots, R_i\}$	Se pierde una entidad que no pertenece a alguna de las subclases	Bueno	Bueno	Redundancia para atributos de entidades que pertenecen a varias subclases	Bueno
5c	$R_p$	Discriminante is null para ocurrencias de entidades que no pertenecen a ninguna subclase		Bueno		Muchos nulos (*)
5d	$R_p$				Bueno	Muchos nulos (*)

(\*) Los pasos 5c y 5d son recomendables cuando hay pocas subclases y éstas tiene pocos atributos.

### 5 Caso: Ejemplo EE/R y mapeo a RM/B

A continuación se presenta el EE/R de un modelo simplificado que recoge los vinos que elabora cada bodega, su composición por tipos de uva y clasificados en jóvenes o de crianza. Cuando se trate de vinos de crianza se registrará la calidad de las distintas añadas.



El mapeo a RM/B quedaría:

Relaciones	PK	FK
<b>Bodegas</b> (Bodega, DO)	Bodega	
<b>Teléfonos</b> (Bodega, Teléfono)	(Bodega, Teléfono)	Bodega/ <b>Bodegas</b>
<b>Vinos</b> (Vino, Grados, Bodega, Categoría <sup>2</sup> , Tiempo_ envejecimiento)	Vino	Bodega/ <b>Bodegas</b>
<b>Uvas</b> (Uva)	Uva	
<b>Compone</b> (Vino,Uva,Porcentaje)	(Vino,Uva)	Vino / <b>Vinos</b> Uva/ <b>Uvas</b>
<b>Añadas</b> (Vino, Año, Calidad)	(Vino, Año)	Vino/ <b>Vinos</b>

Para el mapeo de la jerarquía se ha optado por la opción 5c ya que hay sólo dos subclases y con pocos atributos.

<sup>2</sup> Categoría toma los valores de los subtipos del vino, es decir “joven” o “crianza” para discriminar la pertenencia a un tipo u otro.